UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Diseño de una sistemática de análisis de posicionamiento web

Autor: Adrián González Saiz
Tutor: Eloy Anguiano Rey

marzo 2019

Adrián González Saiz
Diseño de una sistemática de análisis de posicionamiento web

Adrián González Saiz
C\ Francisco Tomás y Valiente Nº 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

Agradecimientos

Tengo mucho que agradecer a muchas personas en lo relativo a la realización de este TFG, pero sin duda alguna comienzo por mi padre, que aunque ya no esté, me motivó tantísimo para terminar la carrera y tantos otros proyectos que he emprendido por mi cuenta.

Agradecimientos también a toda mi familia, a mi novia y mis amigos por estar siempre ahí, incluso en los momentos en que la masiva carga de estudios y emprendimientos me desbordaba.

Por supuesto, también tengo que agradecer mucho a Eloy, por proporcionarme esta completa plantilla en LaTeX 2$_\varepsilon$, el tiempo que me ha dedicado para poder realizar el TFG, resolver mis dudas, y completar el planteamiento inicial del proyecto.

Gracias también a Álvaro Saez, del blog chuiso.com, gran experto en todo lo relacionado con SEO, White Hat SEO y Blackhat SEO que acumula muchos años de experiencia en el sector del posicionamiento web y me indicó que factores son mejores para comenzar un buen análisis, así como los programas de pago que mayor cantidad de información de calidad devuelven y de mayor cantidad de palabras clave, para un mejor análisis.

Resumen

Posicionar un sitio web en el gran buscador es hoy en día mucho más complicado que posicionarlo en buscadores menos desarrollados como Yandex o Bing, ya que Google ha estado actualizando su familia de algoritmos (PageRank) desde finales de los 90, a diferencia de las otras dos, mucho más recientes y con menos presupuesto.

Manipular un algoritmo básico de búsqueda es ridículamente sencillo con unos mínimos conocimientos de seo onpage y seo offpage, ya que basta con crear o comprar unos cuantos enlaces en sitios web de gran reputación o repetir una gran cantidad de veces la palabra clave por la que deseamos que aparezca posicionada nuestra web y ya habremos logrado engañar al algoritmo, pero... ¿realmente sirve esto para manipular el algoritmo de Google hoy en día?... rotundamente no.

Esta respuesta tan contundente motiva el diseño de la sistemática de análisis propuesta para aproximarnos de modo muy preciso a los principales factores de posicionamiento en el buscador de Google, los cuales obviamente no pueden ser deducidos simplemente con un par de búsquedas “a ojo”, motivo por el cual será necesario emplear técnicas de big data para “filtrar el ruido” y sacar conclusiones que nos permitan saber qué hacer para lograr una web de éxito que posiciona por gran cantidad de palabras clave en las SERPS.

Palabras clave

SEO, Google penguin, Google panda, Big data, Query, Scraper, API, Proxy, Gradient Boosting, Tree, Extra Trees, Random Forest

Abstract

Ranking a website in the big search engine is nowadays much more complicated than ranking it in less developed search engines like Yandex or Bing, as Google has been updating its family of algorithms (PageRank) since the late 90's, unlike the other two, much more recent and with less budget.

Manipulating a basic search algorithm is ridiculously simple with a minimum knowledge of SEO On Page and SEO Off page, since it is enough to create or buy a few links in reputable websites or repeat a lot of times the keyword which we want our website to appear positioned, so we have already managed to deceive the algorithm, but... does this really work for manipulating Google's algorithm today? Absolutely not.

This overwhelming response motivates the design of the analysis system proposed to approach, in a very precise way, to the main ranking factors in the Google search engine, which obviously cannot be deduced simply with a couple of searches, the reason why it will be necessary to use big data techniques to filter the noise and draw conclusions that allow us to know what to do to achieve a successful website that ranks by a large amount of keywords in the SERPS.

Keywords

SEO, SERPS, SEM, Google penguin, Google panda, Big data, Query, scraper, API, Proxy, Gradient Boosting, Tree, Extra Trees, Random Forest

ÍNDICE

1 Introducción **1**

1.1 Motivación 2

1.2 Objetivos 3

2 Estado del arte **5**

2.1 Técnicas existentes para diseñar una sistemática de análisis de posicionamiento web 5

2.1.1 Técnicas existentes para extraer parámetros de URL en las SERPS de Google . . . 6

2.1.2 Técnicas existentes para analizar los datos almacenados en la hoja de cálculo o base de datos 6

2.2 Elección de la herramienta SEOquake 8

2.3 Análisis Big Data realizado en Python 9

3 Diseño y arquitectura **11**

3.1 Creación del diccionario de palabras clave 11

3.2 Obtención de los parámetros de posicionamiento en SEOquake 13

4 Integración, pruebas y resultados **19**

4.1 Carga de los datos obtenidos en la herramienta SEOquake 19

4.2 Clasificación de los datos obtenidos de SEOquake 21

4.3 Análisis de los algoritmos 22

4.3.1 Tree 22

4.3.2 Extra Trees 23

4.3.3 Random Forest 23

4.3.4 Gradient Boosting 24

4.4 Procedimiento para clasificar con cada algoritmo 25

4.5 Comparativa entre algoritmos 26

4.6 Acciones a realizar en base al análisis realizado 28

5 Conclusiones y trabajo futuro **29**

5.1 Conclusiones 29

5.2 Trabajo futuro 30

Bibliografía **32**

Definiciones **33**

Acrónimos **35**

Listas

Lista de códigos

4.1 Carga de datos CSV . . . 19
4.2 Eliminar filas del CSV . . . 19
4.3 Adecuar formato de Likes en Facebook . . . 20
4.4 Adecuar formato de fechas de Webarchive . . . 20
4.5 Filtrado de valores NaN o vacíos . . . 21
4.6 Creación de la columna Top Ranked . . . 21
4.7 Selección de los parámetros para la clasificación . . . 25
4.8 Creación del modelo a utilizar . . . 25
4.9 Entrenamiento del modelo a utilizar . . . 25
4.10 Cálculo de la importancia de atributos y porcentaje de aciertos . . . 26
4.11 Validacion cruzada con Gradient Boosting . . . 26

Lista de figuras

3.1 Flujo de información . . . 11
3.2 Parámetros empleados . . . 14
3.3 Funcionamiento de SEOquake . . . 15
3.4 CSV Obtenido . . . 16

Lista de tablas

4.1 Primera tabla comparativa de algoritmos . . . 26
4.2 Segunda tabla comparativa de algoritmos . . . 27
4.3 Tercera tabla comparativa de algoritmos . . . 27

1

Introducción

Hoy en día, posicionar una página web no es trivial, ya que el algoritmo de posicionamiento de Google ha evolucionado en gran medida desde finales de los 90 [1] a causa de los constantes intentos de manipulación que ha sufrido por parte de webmasters de todo el mundo.

Es por ello que hoy en día hay infinidad de factores [2] que determinan la calidad de un sitio web y por tanto su posicionamiento en el buscador de Google, ya que cuantos más factores tenga en cuenta el algoritmo, más complicado será manipular los rankings de una web.

Por si alguien se ha hecho la pregunta de si hoy en día se siguen pudiendo manipular los rankings de las webs en el buscador de Google. . . si, sigue siendo posible, pero de manera mucho más compleja que hace 10 años.

Este es el *leitmotiv* de este trabajo de fin de grado, nacido de mi pasión por crear webs y posicionarlas en el buscador de Google. Se realizarán experimentos en diferentes páginas web y se observarán resultados con herramientas que monitorizan las principales palabras clave por las que un sitio web se encuentra posicionado en el buscador de Google.

El objetivo de todo webmaster es lograr posicionar su página web en el buscador de Google, ya que así logra rendimiento económico, fama, o simplemente compartir su afición o vivencias personales con los visitantes de su página web. Nada de esto ocurre sin lograr aparecer en las primeras posiciones de la primera página del buscador. Este handicap motiva el trabajo de fin de grado.

Lo que se pretende crear en este trabajo de fin de grado es una sistemática que indique cuales son los factores más determinantes que permitan determinar los factores más relevantes a la hora de posicionar una página web en el buscador de Google empleando técnicas de big data y los resultados obtenidos de herramientas profesionales con el objetivo de lograr que nuestra página web incremente notablemente su número de visitantes procedentes del buscador de Google de forma consistente en el tiempo.

En este punto es necesario preguntarse entonces que es lo que tiene en cuenta el algoritmo de Google para mostrar un sitio web.

En primer lugar es necesario saber que una web aparece en Google en una posición diferente para

cada palabra clave, llamada también query. Para cada palabra clave, como podría ser “como ganar dinero en internet”, un sitio web estará posicionado en un lugar diferente del buscador, como podría ser por ejemplo la posición 5 (Página 1, posición 5). Cada página alberga 10 resultados diferentes de media, aunque en algunas búsquedas Google muestra de 8 a 14 resultados en la primera página.

El algoritmo de Google tiene en cuenta la serie de factores anteriormente mencionados para cada palabra clave a buscar, de forma que si se desea posicionar cierto sitio web para la palabra clave “como ganar dinero en internet”, será necesario emprender las acciones adecuadas en la página web en cuestión en base a esa palabra clave, asociada a una URL en concreto, como podría ser por ejemplo la hipotética URL [1], que contiene un artículo de título: “Como Ganar Dinero en Internet en 3 Sencillos Pasos”.

Teniendo ya una mejor idea de lo que se desea conseguir para posicionar un sitio web en el buscador de Google, se puede comenzar a hablar de objetivos, los siguientes pasos necesarios para filtrar los factores de posicionamiento y el análisis de los datos obtenidos.

1.1. Motivación

La motivación de este trabajo de fin de grado es, como ya se ha comentado diseñar un sistemática de análisis de posicionamiento en el buscador de Google que permita saber en todo momento que hacer para lograr que cierta página web ocupe las primeras posiciones del gran buscador para todas las palabras clave posibles, de manera consistente en el tiempo.

Para ello, se emplearán varios datos de cada URL posicionada para cada una de las aproximadamente 1000 palabras clave que servirán de diccionario de partida, de forma que empleando técnicas de Big Data se pueden obtener muy buenas conclusiones, obteniendo patrones y llegando a conclusiones muy realistas.

Además, se compararán los factores de posicionamiento que supuestamente según las webs especializadas en posicionamiento web logran que una web se posicione en un buscador frente a los factores que realmente posicionan una web según el análisis de los datos obtenidos con la herramienta comentada más adelante, y tras filtrar estos con Big Data.

En síntesis, **la motivación de este proyecto es descubrir empíricamente que es lo que debe hacer un webmaster para lograr los mejores posicionamientos en el buscador de Google** y, por tanto, los mayores beneficios económicos de su sitio web.

[1] https://monelandia.com/como-ganar-dinero-en-internet-en-3-sencillos-pasos

1.2. Objetivos

Los objetivos de este trabajo de fin de grado son los siguientes:

O-1.– Realizar un diccionario de 1000 palabras clave lo más realista posible para no interferir con palabras clave de alta competencia [3] como pueden ser “como bajar de peso”, “como ganar dinero”, “bitcoin”, etc. Teniendo en cuenta que una persona utiliza de media de 1000 palabras en su día a día [4], un diccionario de 1000 palabras clave parece más que razonable y objetivo para el proyecto.

O-2.– Emplear la herramienta SEOquake [5] para escanear las primeras posiciones para cada palabra clave del diccionario, de la primera página del buscador de Google, ya que las primeras posiciones siempre son las más significativas, y exportar a una hoja de cálculo los resultados obtenidos.

O-3.– Recuperar en Python los datos volcados por la herramienta a la hoja de cálculo, clasificar los datos con algoritmos como Gradient Boosting, Tree, Extra Trees y Random Forest hasta obtener el mejor resultado, y observar cuales son los factores de posicionamiento más comunes entre las URL posicionadas para cada palabra clave del diccionario.

O-4.– Analizar los resultados obtenidos en la clasificación en Python con el algoritmo ganador y concluir cuales han sido los factores determinantes a la hora de permitir que una URL aparezca en las mejores posiciones para una palabra clave cualquiera.

O-5.– Comentar si realmente tienen sentido los resultados obtenidos, comparándolos con las principales teorías y experimentos actuales.

Desde un principio, la premisa fundamental del proyecto ha sido simplemente descubrir que es lo que hay que hacer para poder posicionar un sitio web, diseñando una sistemática consistente, ya que no es viable emprender más de 200 acciones diferentes para cada URL con el fin de posicionar una página web para una palabra clave.

Una vez explicada la finalidad del proyecto en cuestión y por qué interesa saber qué hacer para posicionar una web y de que manera, lo más importante será dar detalles de los pasos seguidos para obtener los factores de posicionamiento con la herramienta y el análisis final con el python desarrollado para filtrar la información y llegar a la conclusión final.

2

Estado del arte

En este apartado se explicarán las técnicas aplicadas para poder realizar el correcto tratamiento de los datos obtenidos de la herramienta de análisis de las SERPS, así como las técnicas que se están empleando hoy en día para extraer parámetros valiosos en las SERPS para determinar cuales son los más relevantes para el algoritmo de posicionamiento del buscador de Google.

En primer lugar, está el **análisis de la herramienta SEOquake**, que hace posible obtener una buena muestra de las SERPS para obtener los datos en bruto, que lograrán conformar el diccionario de unas 1000 palabras clave, que representan la mayoría de palabras empleadas por un español durante un día.

En segundo lugar, está el **análisis realizado en Python**, el cual recogerá el análisis de la hoja de cálculo, almacenando todos los campos y clasificando los datos para analizar las SERPS de las aproximadamente 1000 palabras clave.

2.1. Técnicas existentes para diseñar una sistemática de análisis de posicionamiento web

Cualquier intento de diseñar una sistemática de análisis de posicionamiento web requerirá poder extraer los parámetros de posicionamiento de las URL que aparezcan en el TOP 10 del buscador de Google para cada una de las palabras que conformen el diccionario de palabras clave elegido.

Esto es así porque es necesario contar con unos datos de partida para poder realizar el posterior análisis de dichos datos, que serán analizados mediante las adecuadas técnicas Big Data, empleando diferentes algoritmos que resultarán mas o menos adecuados en función de los porcentajes de acierto que logren clasificando los datos.

2.1.1. Técnicas existentes para extraer parámetros de URL en las SERPS de Google

Sea cual sea la técnica a emplear para extraer los parámetros de las URL en cuestión, será necesario emplear una API para consultar cada parámetro de cada URL.

Esto es así porque no sería inteligente implementar un potente scraper junto con una base de datos exageradamente grande para obtener información que ya facilitan servicios como Alexa y Semrush, entre otros. Se asume entonces que será necesario emplear una API por cada parámetro de cada URL.

Existen varias técnicas de extracción de los parámetros más importantes de las URL obtenidas, las cuales son:

- Consultar uno a uno los parámetros de cada URL del TOP 10 manualmente en cada servicio: Sin duda es la peor técnica, ya que el tiempo que tomaría conformar una hoja de cálculo o base de datos es desorbitado, ya que es necesario un gran número de palabras clave, aún mayor número de URL, y aún mayor número de parámetros.
- Emplear una herramienta que obtenga todos los parámetros de cada URL del Top 10, realizando las peticiones pertinentes a cada API asociada a cierto parámetro: Esta ha sido la técnica elegida para el presente trabajo de fin de grado, ya que permite conformar un diccionario de 1000 palabras clave con relativa facilidad y rapidez. Esta técnica presenta el inconveniente de que es necesario introducir en el buscador de Google cada palabra clave del diccionario y exportar los datos obtenidos a una hoja de cálculo, uniendo todos los datos resultantes en una única hoja de cálculo. Aunque pueda parecer tedioso introducir cientos de palabras clave en el buscador de Google y exportar cientos de hojas de cálculo, resulta menos tedioso que diseñar una herramienta que amplie las funcionalidades de la herramienta SEOQuake, la elegida en este trabajo de fin de grado, logrando realizar cientos o miles de consultas al buscador de Google y a las API asociadas a cada parámetro.
- Diseñar una herramienta capaz de realizar cientos o miles de consultas al buscador de Google y a las APIs de los servicios asociados a cada uno de los parámetros de cada URL del TOP 10 de cada una de las palabras clave: Esta tarea por si sola podría ser la piedra angular de un trabajo de fin de grado, ya que esta herramienta necesitaría implementar proxys para evadir los controles anti-spam del buscador de Google, scrapers, y comunicarse con herramientas cuyo acceso via API supone un coste económico elevado. Hoy en día no existe una herramienta capaz de realizar esta tarea, por lo que esta opción tan novedosa y eficiente queda descartada.

2.1.2. Técnicas existentes para analizar los datos almacenados en la hoja de cálculo o base de datos

En este punto ya es necesario hablar de tratamiento de datos y de técnicas Big Data. Se comenzará por exponer las principales aplicaciones del Big Data y posteriormente se profundizará en los distintos algoritmos o técnicas disponibles para llegar a conclusiones en base a miles de datos almacenados en una base de datos.

Hoy en día este tipo de técnicas son ampliamente utilizadas para infinidad de tareas [6], pero sus aplicaciones principales son las siguientes:

Ámbito de la salud: Se han puesto muy de moda las pulseras que monitorizan las constantes vitales. Estas pulseras exportan los datos producto del monitoreo durante el ejercicio y además de comparar las pulsaciones entre usuarios, pueden determinar el estado de salud del individuo.

Predicciones: Gracias al Big Data es posible realizar predicciones en base a datos obtenidos. Las entidades bancarias aplican predicciones basadas en Big Data continuamente, especialmente para conceder créditos e hipotecas, ya que necesitan conocer al cliente para realizar la pertinente evaluación de riesgo.

Economización en los medios de transporte: También es posible emplear técnicas Big Data para reconocer rutas ineficientes empleadas por medios de transporte. Gracias al pertinente análisis de datos, las empresas propietarias de estos medios de transporte pueden tomar decisiones en base a estos datos.

Técnicas Big Data más empleadas

En primer lugar, es necesario aclarar cuales son los lenguajes de programación que más se adaptan a estas técnicas Big Data, haciendo posible emplear algoritmos Machine Learning para clasificar los datos obtenidos.

En este trabajo de fin de grado se ha empleado el lenguaje de programación Python, ya que es un lenguaje muy potente, simple, y cuenta con gran cantidad de librerías. No obstante, existen 6 [7] más que también podrían haberse empleado, los cuales son:

- R
- LISP
- PROLOG
- JAVA
- C++
- TORCH

Una vez elegido el lenguaje de programación [8], es necesario tener en cuenta los principales algoritmos que podemos emplear en Python. Aunque como se explicará más adelante existen variaciones y mas tipos concretos, los principales algoritmos machine learning son los siguientes:

Árboles de decisión - Aprendizaje supervisado [9]: Los árboles de decisión son una herramienta de apoyo a la decisión que utiliza un gráfico o un modelo similar a un árbol de decisiones y sus posibles consecuencias, incluidos resultados de eventos fortuitos, costes de recursos y utilidad.

Naïve Bayes - Aprendizaje supervisado: Estos clasificadores son una familia de clasificadores probabilísticos sencillos basados en la aplicación de Bayes, concretamente en el

teorema con fuertes supuestos de independencia entre las características de los atributos entre sí.

Regresión Lineal - Aprendizaje supervisado: Este método se emplea para realizar regresión lineal. La regresión lineal es similar a ajustar una línea recta a través de un conjunto de puntos.

Regresión Logística - Aprendizaje supervisado: La regresión logística es una manera estadística de modelar un resultado binomial con una o más variables explicativas. Este método mide la relación entre la variable dependiente categórica y una o más variables independientes estimando las probabilidades utilizando una función logística.

SVM - Aprendizaje supervisado: También llamado Support Vector Machines, es un algoritmo de clasificación binario. Dado un conjunto de puntos de 2 tipos en el lugar N dimensional, SVM genera un hiperplano dimensional para separar esos puntos en 2 grupos.

Métodos Ensemble - Aprendizaje supervisado: Los métodos ensemble son algoritmos de aprendizaje que construyen un conjunto de clasificadores para posteriormente clasificar nuevos puntos de datos tomando un voto ponderado de sus predicciones.

Algoritmos Clustering - Aprendizaje no supervisado: Los algoritmos clustering agrupan un conjunto de objetos tales que los objetos en el mismo grupo (cluster) son más similares entre sí que a los de otros grupos, formando clusters y clasificando así la información.

PCA - Aprendizaje no supervisado: También llamado Análisis de Componentes Principales, es un procedimiento estadístico que emplea una transformación ortogonal para convertir un conjunto de observaciones de variables posiblemente correlacionadas en un conjunto de valores de variables linealmente no correlacionadas, las llamadas componentes principales.

SVD - Aprendizaje no supervisado: También conocido como Singular Value Decomposition, en el álgebra lineal es una factorización de una matriz compleja real.

ICA - Aprendizaje no supervisado: También llamado Análisis de Componentes Independientes, es una técnica estadística empleada para revelar los factores ocultos que subyacen a conjuntos de variables, mediciones o señales aleatorias.

2.2. Elección de la herramienta SEOquake

Tras una comparativa que será explicada en detalle en el apartado de diseño, se decidió emplear la herramienta SEOquake, ya que es la que mayor cantidad de parámetros relevantes muestra de cada URL del TOP 10 de las principales herramientas de análisis de SERPS. Al ya existir este tipo de herramientas que extraen parámetros, no se ha planteado la posibilidad de desarrollar una herramienta así, ya que solo esa tarea justificaría la realización de otro TFG aparte, al ser necesario desarrollar un

complejo scraper con acceso a API para consultar los parámetros de las SERPS y servidores proxy para evitar los controles anti spam del buscador de Google.

Respecto a las 1000 palabras clave elegidas, se ha optado por ellas como semilla porque al ser palabras tan genéricas disminuye significativamente la posibilidad de encontrar resultados alterados [10]. Entiéndase por resultados alterados una palabra clave de gran competencia como por ejemplo "como ganar dinero" por la cual todas las webs desean aparecer y, por tanto, pueden emprender acciones muy efectivas pero poco duraderas en el tiempo que obviamente el algoritmo de Google penalizará en cuestión de semanas al ejecutar Google Penguin o Google Panda. No es interesante hacer una captura de unas SERPS que van a variar en muy poco tiempo porque esos datos tendrían poca fiabilidad, por lo tanto no son palabras clave representativas.

2.3. Análisis Big Data realizado en Python

Tras realizar pruebas con diferentes algoritmos, finalmente se llegó a la conclusión de que con estos datos en particular, los algoritmos que mejor clasifican los datos son los siguientes:

Extra Trees: Este algoritmo cuya documentación puede leerse en https://scikit-learn.org implementa un metaestimulador que se ajusta a un número de árboles de decisión aleatorios llamados árboles extra en varias submuestras del conjunto de datos y utiliza el promediado para mejorar la precisión predictiva y controlar el sobreajuste.

Random Forest: El algoritmo Random Forest, cuya documentación puede leerse en https://scikit-learn.org es un metaestimulador que se ajusta a un número de clasificadores de árboles de decisión en varias submuestras del conjunto de datos y utiliza el promedio para mejorar la precisión predictiva y el ajuste excesivo de control.

Gradient Boosting: Este algoritmo, cuya documentación puede leerse en https://scikit-learn.org construye un modelo aditivo en una etapa avanzada, permite la optimización de funciones de pérdida arbitrarias y diferenciables.

Todos ellos se encuentran en https://scikit-learn.org, una librería de Python empleada en minería y análisis de datos basada en NumPy, SciPy y matplotlib.

El análisis de las 1000 palabras semilla mostrará cuales son los parámetros de los brindados por la herramienta SEOquake decisivos a la hora de posicionar una web. Para poder visualizar correctamente los resultados del análisis, se realizará una tabla a fin de detectar patrones, los cuales concluirán el análisis.

3

Diseño y arquitectura

Como se mencionó anteriormente, el diseño de la sistemática de análisis de posicionamiento web se basa en dos fases destacadas. La primera es la obtención de los parámetros de posicionamiento de la primera página del buscador de Google España (Google.es) para cada una de las 1000 palabras clave semilla, las cuales fueron elegidas por ser representativas del vocabulario habitual diario de un español. Todos esos datos serán volcados a un documento Excel. La segunda fase consiste en analizar esos datos en Python usando técnicas de Big Data. Se detallarán dichas técnicas durante este capítulo.

En la figura 3.1 se puede ver el flujo de información de forma más gráfica:

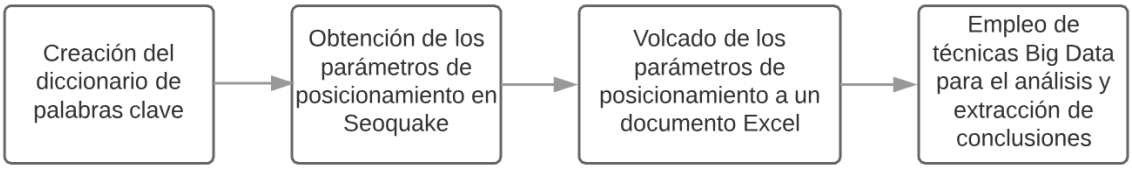

Figura 3.1: En primer lugar se crea el diccionario de palabras clave a partir del cual se realizará un análisis de las SERPS para posteriormente obtener los parámetros de posicionamiento de cada URL del TOP 10 en el buscador de Google. Tras realizar ese proceso, se exportan a excel los datos de posicionamiento obtenidos mediante la herramienta SEOQuake para analizarlos con técnicas Big Data y poder así concluir cuales son los parámetros más relevantes a la hora de posicionar una web.

A continuación, se explica en detalle cada una de las 4 etapas, formando parte de la fase 1 las tres primeras y la fase dos que es la última etapa.

3.1. Creación del diccionario de palabras clave

Sin diccionario de palabras clave no tiene razón de ser este trabajo de fin de grado. Aunque quizás haya sido la etapa más trivial de todas ellas, resulta de vital importancia la creación de un diccionario de palabras clave representativo del vocabulario diario de un español. Esto elevará notablemente las posibilidades de realizar un análisis correcto, puesto que es necesario obtener los parámetros de posicionamiento de URL posicionadas únicamente para palabras clave que no sean objeto de la

competencia desmedida de ciertas empresas y/o webmasters que podrían emplear ciertas técnicas blackhat SEO que introducirían ruido en nuestro análisis. Esto es así porque algunas de esas URL que conforman la primera página del buscador de Google estarían empleando técnicas poco legítimas que no perdurarán en el tiempo, y posteriormente ocuparán su lugar otras URL que emplean técnicas whitehat SEO [11].

Descartando entonces del diccionario de palabras clave ciertas palabras que añadirían ruido indeseado como podrían ser “como ganar dinero en internet”, “como bajar de peso”, etc. se opta entonces por conformar un diccionario que pretenda resumir el vocabulario diario de un hispanohablante, ya que así se eliminan estas palabras clave no representativas y a la vez se conforma una muestra fiel a la realidad, limpia de resultados alterados en las SERPS.

Respecto al tamaño del diccionario de palabras clave, no se ha elegido una cantidad de 1000 por casualidad. Como se comprobó investigando [4], una persona adulta sin estudios universitarios emplea unas 1000 palabras en su día a día, mientras que una persona adulta con estudios universitarios, unas 3000. Se considera así por tanto, que un diccionario de unas 1000 palabras aproximadamente es suficiente. Es por ello que la fuente de palabras clave elegida ha sido el CREA, ya que cuenta con los datos más rigurosos, incluyendo las frecuencias absolutas y normalizadas de cada una de las 1000 palabras clave.

La página web de la RAE proporciona incluso más de las 10.000 palabras más utilizadas, como se puede observar en el corpus de la RAE [1].

Por tanto, una vez obtenido el documento de texto con las 1000 palabras más empleadas, el cual se puede ver también en el corpus de la RAE el siguiente paso fue modificar ese documento de texto, ya que para introducir esas 1000 palabras clave en la herramienta para obtener los parámetros de la primera página de las URL en el buscador de Google sobran los números, puntos, espacios, y todo tipo de impurezas.

Es en este paso cuando las expresiones regulares resultan de gran utilidad, ya que es totalmente inviable realizar este filtrado manualmente. Una opción sería realizar un sencillo programa empleando Flex y C para realizar este tratamiento del documento de texto, pero es una tarea tan trivial que podía ser llevada a cabo sin problemas con la herramienta online [2], que realiza esta misma tarea indicándole la expresión regular correcta.

En este caso, basta con la expresión regular [0-9]|[.]|[,]|\t|\r para realizar el filtrado ya comentado, logrando eliminar números, puntos, comas, saltos de línea y retornos de carro.

[1] http://corpus.rae.es/lfrecuencias.html

[2] https://www.regexpal.com/

3.2. Obtención de los parámetros de posicionamiento en SEOquake

Como se comentó anteriormente, antes de elegir la herramienta SEOquake, se realizó un análisis de las herramientas disponibles en el mercado, tanto gratuitas como de pago.

El objetivo de la herramienta elegida era claro: analizar la primera página del buscador de Google.es (Google España) para cada una de las 1000 palabras clave obtenidas del CREA y extraer los parámetros de posicionamiento de cada una de sus URL, para analizar posteriormente en Python dichos parámetros y obtener patrones comunes.

Las otras herramientas analizadas para extraer los parámetros de posicionamiento fueron las siguientes:

Semrush: Herramienta relativamente cara y bastante compleja, pero resultó ser de utilidad únicamente si se tiene una página web y se desean realizar reportes de las páginas web de otras personas que se encuentran en posiciones cercanas. Semrush proporciona bastantes datos relativos a popularidad de una página web en redes sociales y enlaces apuntando a dicha página web [12], pero nada más, y lo que se busca es tener una visión global del desempeño de una página web, analizando los parámetros más representativos.

Serpwoo: En un principio era la herramienta elegida para llevar a cabo esta labor, pero tras probar la versión Premium de la herramienta durante varios días se llegó a la conclusión de que presentaba el mismo problema que Semrush y prácticamente todas las herramientas del mercado, ya que únicamente servía para analizar la competencia de una sola página web [13], prohibiendo cualquier opción de analizar todas las URL de la primera página de Google, limitando a un solo resultado y, por tanto, proporcionando un resultado totalmente inútil y para nada representativo. Los parámetros obtenidos en la herramienta Serpwoo eran prácticamente los mismos que en la herramienta Semrush, pero empleando sus propias métricas para obtener parámetros como los enlaces que apuntan a una web.

Sistrix: Sin duda alguna una de las mejores herramientas del profesional en análisis de posicionamiento web actualmente, siendo una herramienta de referencia en el SEO. Aunque la herramienta Sistrix cuenta con una considerable base de datos y una gran variedad de complementos que reportan muchísima información de valor, presenta exactamente el mismo problema que las dos anteriores, ya que solo sirve para estudiar una página web en profundidad, no para diseñar una sistemática de esta naturaleza, de forma que nuevamente limita a una sola página web creada como nuevo proyecto en la herramienta Sistrix.

Sin ninguna duda la herramienta SEOquake, de relativamente poca utilidad para el SEO profesional, resultó ser la herramienta más adecuada para extraer los datos necesarios, ya que cumple varios

requisitos fundamentales, además de ser gratuita:

- SEOquake **permite analizar en base a palabra clave** y no a página web, lo que permite analizar toda la primera página de Google.es introduciendo como query nuestra palabra clave.
- SEOquake cuenta con diversos parámetros que, aunque no sean muy numerosos, son muy interesantes, ya que resumen a la perfección el desempeño de la SEO analizada según los estándares actuales en posicionamiento web.
- SEOquake permite filtrar los parámetros obtenidos de cada SEO, de forma que pude eliminar algunos parámetros que no aportaban nada relevante y solo introducirían ruido en el posterior análisis.

Respecto a los parámetros empleados en el análisis, aquí se puede observar la interfaz de SEOquake para elegir los parámetros que desean introducir en el análisis. Cabe destacar que mientras casi todas las herramientas de este tipo son aplicaciones web o programas ejecutables, SEOquake es una extensión del navegador Google Chrome que se integra perfectamente en la búsqueda en Google.es, contando con varias API para acceder a servicios como Alexa para obtener el ranking Alexa de cada URL, a Webarchive para conocer la antigüedad del dominio, o a Semrush para conocer tan dispares datos como por ejemplo tráfico estimado de la web, precio de ese tráfico, y enlaces a esa URL concreta y al dominio. En conclusión, se dispone con esta herramienta de un scraper gratuito y con acceso a multitud de API.

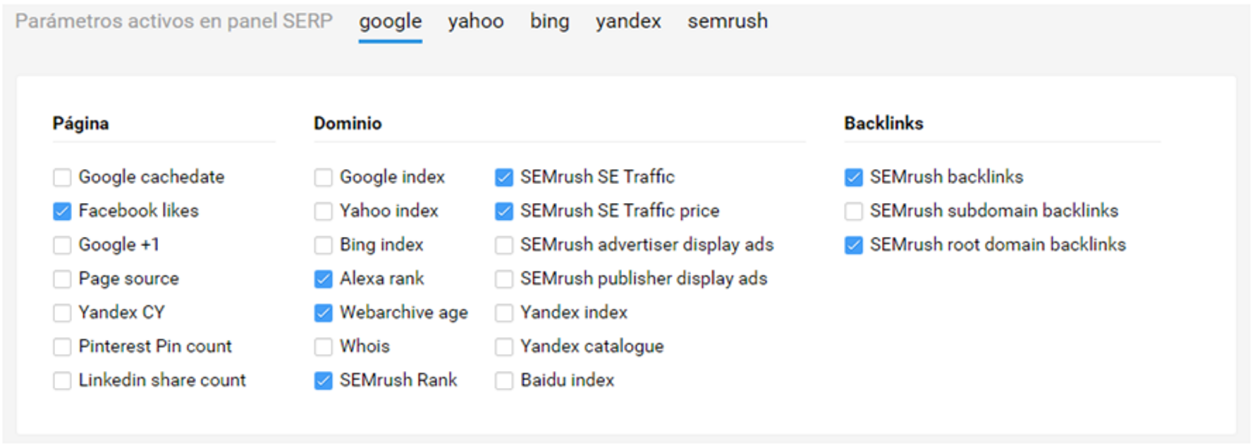

Figura 3.2: Se han elegido los parámetros marcados en color azul por ser los que más cantidad de información aportan. Seleccionando estos parámetros en la herramienta, obtenemos un completo reporte en Excel para cada URL del TOP 10 de cada palabra clave del diccionario.

Como se puede observar, se pueden obtener estos parámetros también en los buscadores Yahoo, Bing, y el ruso Yandex, además de en la popular herramienta Semrush, de la cual toma varios parámetros de gran valor, pero solo resulta de interés el estudio en Google.es, ya que como se ha comentado anteriormente, este es el único buscador lo suficientemente evolucionado y con la suficiente cuota de mercado [14] como para que el estudio resulte de interés.

Como se estudió en la asignatura de inteligencia artificial y en fundamentos de aprendizaje automático, **si se tienen en cuenta muchos parámetros se estaría aumentando mucho la varianza**, de

manera que según la estadística un pequeño cambio en los parámetros produce un gran cambio en la salida. Como conclusión, la mejor opción es realizar una pequeña heurística, que en este caso ha resultado ser guiarse con conocimientos teóricos de posicionamiento web y el de grandes profesionales como Chuiso, Luis M Villanueva, Dean Romero, etc. y buscar las herramientas que cuenten con la mayor cantidad de esos parámetros, para lograr guiar el análisis hacia una solución lo más certera posible.

La única sección de configuración que resulta interesante de SEOquake es la mostrada en la figura 3.2, pero también es interesante destacar el uso de la herramienta explicando la imagen que puede observarse en la figura 3.3

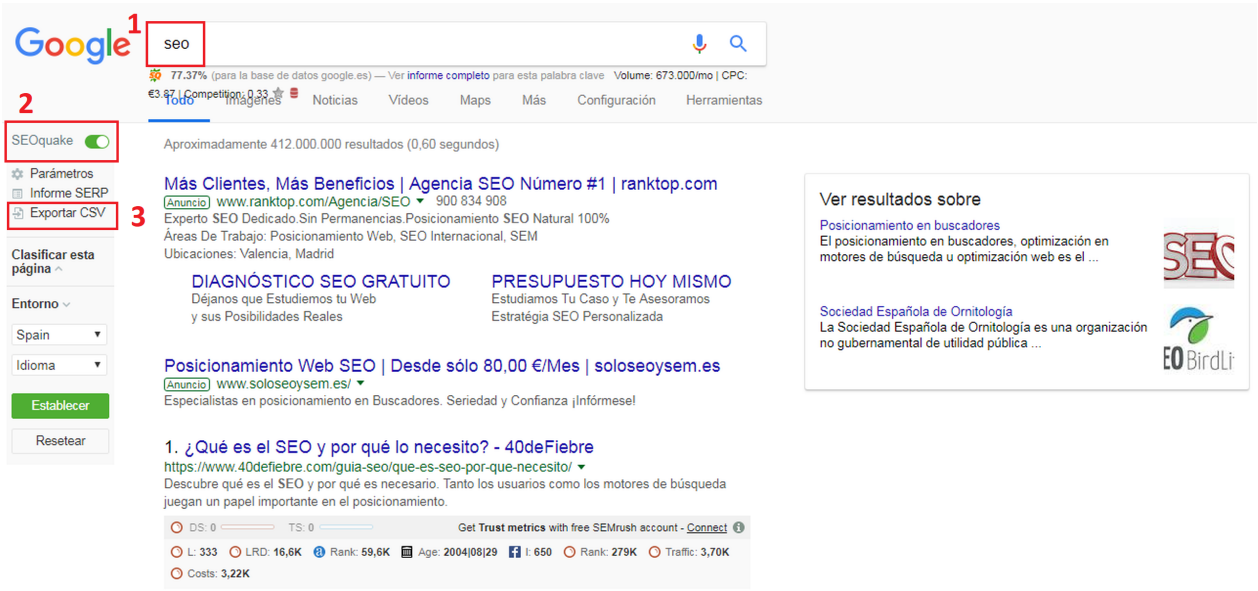

Figura 3.3: Como se puede apreciar en las diferentes secciones de la imagen, introducimos la palabra clave a analizar en SEOQuake (1) en el buscador de Google, ya que se integra en el mismo, para posteriormente activar SEOQuake (2) y exportar el archivo CSV (3).

Cabe destacar que tras configurar los parámetros que se desean ver reflejados en el informe, primero se debe establecer el país España en entorno, ya que así se fuerza a que la heramienta únicamente muestre los resultados de los parámetros en Google.es.

Tras configurar la herramienta, se puede ver como para la palabra clave de ejemplo “seo” se obtiene la lista de URL bajo los anuncios. Bajo cada URL se pueden observar los parámetros de posicionamiento seleccionados anteriormente, los cuales fueron exportados a un Excel en formato .csv, quedando como puede observarse en la figura 3.4

Se puede observar en la figura 3.4 un fragmento de la lista de parámetros de las URL de la primera página para cada una de las 1000 palabras clave en el buscador Google.es.

Los parámetros finalmente elegidos fueron los más relevantes y que aportan mayor cantidad de información, los cuales se explican a continuación:

	A	B	C	D	E	F	G	H	I	J	K
1	#	Url	SEMrush bac	SEMrush roo	Alexa rank	Webarchive	Facebook lik	SEMrush Ran	SEMrush SE T	SEMrush SE	Traffic price
2	de										
3	1	https://polit	0	93618311	410	2011|05|18	2133	1851	1409032	851370	
4	2	https://polit	0	93618311	410	2011|05|18	4555	1851	1409032	851370	
5	3	http://www.	0	15364866	631	2000|05|10	6484	2784	898784	310306	
6	4	https://as.cc	0	21345545	634	2000|02|29	919	2806	892528	575266	
7	5	http://www.	0	7460760	1304	n/a	19	6903	329902	267051	
8	6	https://rsh.r	1533	31191	86314	2015|12|02	7063	1450008	363	6	
9	7	https://www	167	6886650207	2	2005|04|28	522Â mil	2	1570404302	1086433663	
10	8	https://www	1609	2121403	49112	1997|07|02	26	280683	3714	833	
11	9	https://www	4550	13631710	839	2010|08|26	17	181	15225971	29401012	
12	la										
13	1	http://www.	427252	7460760	1304	n/a	31Â mil	6903	329902	267051	
14	2	https://www	3851	1306575	1651	2002|06|03	972	90722	15622	7477	
15	3	https://www	5611	1306575	1651	2002|06|03	20	90722	15622	7477	
16	4	https://www	1373	810987	8721	1999|11|27	399	45640	36201	20407	
17	5	http://www.	0	15364866	631	2000|05|10	6484	2784	898784	310306	
18	6	https://elpai	0	93618311	410	1996|12|19	2266	1851	1409032	851370	
19	7	https://elpai	0	93618311	410	1996|12|19	36	1851	1409032	851370	
20	8	https://elpai	0	93618311	410	1996|12|19	10	1851	1409032	851370	
21	9	https://elpai	0	93618311	410	1996|12|19	16	1851	1409032	851370	
22	que										
23	1	http://www.	772335	1083160	106860	2004|12|02	4787	461499	1915	214	
24	2	http://www.	87	8165463	327	1999|10|13	30	289	9998960	6525792	
25	3	http://www.	0	3049991	1355	1999|10|13	40	48	46653487	34766237	
26	4	https://elpai	0	93618311	410	1996|12|19	383	1851	1409032	851370	
27	5	http://www.	0	7460760	1304	n/a	405	6903	329902	267051	
28	6	http://www.	0	7460760	1304	n/a	2	6903	329902	267051	
29	7	https://as.cc	0	21345545	634	2000|02|29	411	2806	892528	575266	
30	8	https://as.cc	0	21345545	634	2000|02|29	131	2806	892528	575266	
31	9	https://www	0	1937434	6901	2012|05|08	984	65437	23347	17604	

Figura 3.4: En la imagen se puede apreciar el comienzo del CSV obtenido, en el cual se encuentran analizadas las 1000 palabras clave del diccionario. Cada palabra clave aparece acompañada de su TOP 10 en el buscador de Google, con sus respectivos parámetros de posicionamiento.

Facebook Likes: Número de Likes que tiene en Facebook esa URL. Google le da mucha importancia al factor social que aporta Facebook, ya que interpreta que si la gente comparte mucho esa URL en Facebook, significa que la experiencia de usuario de esa URL es aceptable. La teoría dice que cuanto mayor es este parámetro, representa una señal más positiva para la familia de algoritmos del buscador de Google.

Alexa Rank: Ranking de dicha URL en el sitio web de Alexa. La teoría dice que cuanto menor es este parámetro, representa una señal más positiva para la familia de algoritmos del buscador de Google.

Webarchive Age: Antigüedad del dominio de dicha URL. Dicha antigüedad es recuperada via API del sitio web [3]. La teoría dice que cuanto menor es este parámetro, representa una señal más positiva para la familia de algoritmos del buscador de Google, ya que a mayor antigüedad de dominio, mayor confianza a ojos de Google y más extenso historial.

Semrush Rank: Ranking de dicha URL en la popular herramienta Semrush. Obviamente, al igual que ocurre con el ranking en Alexa, cuanto menor sea este parámetro, mayor calidad tendrá el dominio y por tanto mejor tratado deberá ser por la familia de algoritmos del buscador de Google. La teoría dice que cuanto menor es este parámetro, representa una señal más positiva para la familia de algoritmos del buscador de Google.

Semrush SE Traffic: Volumen mensual medio que la herramienta Semrush estima para el dominio al que pertenece dicha URL. La teoría dice que cuanto mayor es este parámetro, representa una señal más positiva para la familia de algoritmos del buscador de Google.

Semrush SE Traffic Price: Según la herramienta Semrush, este parámetro indica el gasto mensual estimado de mostrar anuncios de Google para las palabras clave por las cuales el dominio al que pertenece dicha URL está posicionado en Google. Quizás puede parecer un parámetro absurdo a simple vista, pero la realidad es que Google obtiene el 95 por ciento de sus ganancias [15] a través de su sistema publicitario llamado Adwords, y tiene un complejo algoritmo [16] mediante el cual asigna un CPC a cada dominio y URL perteneciente a ese dominio basándose en multitud de parámetros que motivarían la realización de otro TFG a parte. Semrush SE Traffic Price es un parámetro del que se puede inferir el CPC de dicha URL. Si este parámetro es elevado, significa que este algoritmo empleado por Adwords ha asignado un alto CPC a dicha URL, lo que resulta algo enormemente positivo para la salud de la URL a ojos del buscador de Google. En resumen, el posicionamiento en el buscador de Google de usas URL y la asignación de CPC de los anuncios mostrados en dichas URL está estrechamente relacionado. La teoría dice que cuanto mayor es este parámetro, representa una señal más positiva para la familia de algoritmos del buscador de Google.

Semrush Backlinks: Importantísimo parámetro, representa el número de enlaces que apuntan a esa URL desde otros dominios. Se ha comprobado empíricamente en multitud de

[3] https://web.archive.org

ocasiones que Google le da muchísima importancia a la cantidad, calidad, y tráfico de los enlaces de otras webs hacia la nuestra [17], ya que es una clara muestra de la popularidad de una web. Recientemente se ha comprobado también empíricamente que un enlace en una web A visitado con frecuencia hacia una web B ayuda enormemente al posicionamiento en el buscador de Google de la web B, máxime siendo la web A una web de alta autoridad y tráfico web como podría ser un periódico online [18] como por ejemplo Europa Press. La teoría dice que cuanto mayor es este parámetro, representa una señal más positiva para la familia de algoritmos del buscador de Google.

Semrush Root Domain Backlinks: Se aplica exactamente todo lo comentado en el anterior parámetro, con la salvedad de que este parámetro se refiere únicamente a los enlaces que apuntan al dominio raíz al que pertenece dicha URL. La teoría dice que cuanto mayor es este parámetro, representa una señal más positiva para la familia de algoritmos del buscador de Google.

4

Integración, pruebas y resultados

4.1. Carga de los datos obtenidos en la herramienta SEOquake

En primer lugar se cargan en el programa Python los datos de parametros.csv con la instrucción:

```
data_file = "parametros.csv"
data = pd.read_csv(data_file, sep=";", encoding = "ISO-8859-1")
```

No se tienen en cuenta las palabras para hacer la clasificación, solamente los parámetros y el lugar del ranking en el que quedó la URL correspondiente. Por tanto, se eliminan todas las filas que contienen las palabras, que son las filas que tienen 9 o más valores vacíos:

```
data = data.dropna(thresh=9)
```

Para poder clasificar correctamente los datos es necesario que los datos que se carguen tengan valores numéricos, por lo que se realizan las siguientes correcciones en los datos:

1.– En el parámetro Facebook likes:

Se sustituyen los valores de formato incorrecto generados por la herramienta SEOquake por su correspondiente valor numérico, 10000 ó 10000000 en el ejemplo, y se guarda en un nuevo parámetro llamado Facebook likes number en la figura de Código 4.3

2.– En el parámetro Webarchive age:

Se calcula la diferencia en días de esa fecha respecto a una fecha fija (6 may 2018) y se guarda en un nuevo parámetro Webarchive age days.

Se corrigen también valores de error en ese campo como wait o error generados por el scraper de la herramienta SEOquake y a esos valores se les asigna valor NaN. Se puede observar esto en la figura de Código 4.4

Se borran todos los campos donde al menos uno de los valores de los parámetros es NaN o vacío. Se ha decidido hacer esto así porque agregar otros valores podría generar ruido que interfiera en el algoritmo.

En un momento dado se decidió sustituir esos valores por la media del resto de valores o quizás la mediana, pero quizás se podrían estar añadiendo datos erróneos. También ayudó a tomar esa decisión el hecho de que estos registros representan el 5 por ciento de todos los datos, por lo tanto no es tan grande la pérdida de datos.

```
# Correcciones de formato
def text_to_number(text):
    text = text.replace(',', '.')
    if (text.find("mill")!=-1):
        return float(text.split("")[0]) *1000000
    elif (text.find("mil")!=-1):
        return float(text.split("")[0]) *1000
    else:
        return float(text)

data['Facebook_likes_number'] = data.apply(lambda row: text_to_number(str(row['Facebook_likes'])),
    axis=1)
```

```
# Cálculo del Webarchive age en días
def text_to_date(text):
    # Corrección de formato
    if (text=="wait..." or text=="error"):
        text = "nan"
    # Calcular la diferencia en días entre fecha 6 may 2018 y webarchive age
    if (text!="nan"):
        date_text = text.split("|")
        year = int(date_text[0])
        month = int(date_text[1])
        day = int(date_text[2])
        text = abs(date(2018, 5, 6)-date(year, month, day)).days
    return float(text)

data['Webarchive_age_days'] = data.apply(lambda row: text_to_date(str(row['Webarchive_age'])),
    axis=1)
```

Por eso también se imprime ese dato de porcentaje de registros borrados. Se puede observar el código que realiza el filtrado en la figura de Código 4.5

```
# Eliminar todos los registros donde alguno de sus valores es Nan e imprimir en pantalla
# la cantidad de registros restantes y el porcentaje que se mantuvo tras la eliminación
total_num_rows = data.shape[0]
data = data.dropna(axis=0, how='any')
filt_num_rows = data.shape[0]
per_filt_rows = int(100*(filt_num_rows/total_num_rows))
print('Quedan {0} de {1} filas ({2} %) tras filtrar los valores NaN'.format(filt_num_rows,
    total_num_rows, per_filt_rows))
print('')
```

Por último, se desea que el algoritmo aprenda a clasificar lo siguiente. Dados ciertos valores de parámetros, detectar si esos valores hacen que la URL se posicione en una posición muy alta del ranking. En este caso se ha decidido que esa posición sea posicionar entre los 3 primeros (aunque ese valor se puede cambiar en el código fácilmente). **Se ha creado para eso la nueva columna Top ranked que tiene el valor True si la URL se posiciona entre los 3 primeros y False si no se da el caso.**

```
# Creación de la columna Top ranked, la cual es True si el dato se encuentra entre los 3 primeros
data['#'] = data.apply(lambda row: int(row['#']), axis=1)
data['Top ranked'] = np.zeros(data.shape[0])
data['Top ranked'] = data["#"] < 4
```

4.2. Clasificación de los datos obtenidos de SEOquake

Para clasificar los datos y aprender modelos que sean capaces de predecir rankings, se emplearon estos 3 algoritmos de aprendizaje automático:

1.– Regresión Logística

2.– Árboles

3.– Ensemble de Árboles (ExtraTrees, RandomForest y Gradient Boosting)

Respecto al algoritmo ensemble de árboles, son un conjunto de modelos de árboles que al unirse crean una predicción más fuerte que utilizando los árboles sencillos.

Generalmente, utilizar una unión de diferentes modelos a través de un ensemble logra mejorar los resultados que dan los modelos por separado [19].

Como se comentó anteriormente, empleando los diferentes algoritmos de clasificación de scikit learn se fue testeando como se comportan los datos respecto a los mismos.

En los métodos de aprendizaje de árboles, cabe destacar la función feature_importances_, en la que se puede estimar el peso o importancia de cada parámetro en ese modelo.

4.3. Análisis de los algoritmos

En esta sección se mostrarán los análisis de los algoritmos anteriormente mencionados, comentando los resultados obtenidos sobre el conjunto de datos analizado en el presente trabajo de fin de grado.

4.3.1. Tree

Consultando las referencias de la explicación de estos algoritmos en el capítulo 2 se puede encontrar una buena explicación de la librería scikit learn sobre cada uno de los cuatro algoritmos que se comentarán en este apartado.

Estos son los datos obtenidos mediante el algoritmo Tree:

Importancia según atributos

- SEMrush backlinks: 0.185
- SEMrush root domain backlinks: 0.052
- Alexa rank: 0.233
- SEMrush Rank: 0.044
- SEMrush SE Traffic: 0.05
- SEMrush SE Traffic price: 0.080
- Facebook likes number: 0.223
- Webarchive age days: 0.133

Porcentaje de aciertos en entrenamiento

Empleando el algoritmo Tree, el porcentaje de aciertos en entrenamiento fue del 92.5 por ciento. Esto quiere decir que el algoritmo detecta bien un poco más de 9 de 10 ejemplos de entrenamiento.

Resultados de la validación cruzada

El valor medio de todos los parámetros empleando el algoritmo Tree fue de 0.679, mientras que la desviación estándar fue de 0.019.

Para apreciar el desempeño del algoritmo aprendido se empleó validación cruzada con 10 subconjuntos. Esto consiste en validar un subconjunto, utilizar los 9 subconjuntos restantes para entrenar al modelo y validar a ese modelo sobre ese subconjunto restante. **En este caso el porcentaje de aciertos se reduce bastante.** Aplica la misma explicación al resto de algoritmos.

4.3.2. Extra Trees

Empleando el algoritmo Extra Trees, los datos obtenidos fueron los siguientes:

Importancia según atributos

- SEMrush backlinks: 0.223
- SEMrush root domain backlinks: 0.067
- Alexa rank: 0.057
- SEMrush Rank: 0.059
- SEMrush SE Traffic: 0.127
- SEMrush SE Traffic price: 0.119
- Facebook likes number: 0.267
- Webarchive age days: 0.081

Porcentaje de aciertos en entrenamiento

Empleando el algoritmo Extra Trees, el porcentaje de aciertos en entrenamiento también fue del 92.5 por ciento.

Resultados de la validación cruzada

El valor medio de todos los parámetros empleando el algoritmo Extra Trees fue en esta ocasión de 0.718, mientras que la desviación estándar fue de 0.018.

4.3.3. Random Forest

Empleando el algoritmo Random Forest, los datos obtenidos fueron los siguientes:

Importancia según atributos

- SEMrush backlinks: 0.203
- SEMrush root domain backlinks: 0.072
- Alexa rank: 0.143
- SEMrush Rank: 0.085
- SEMrush SE Traffic: 0.09
- SEMrush SE Traffic price: 0.086
- Facebook likes number: 0.241
- Webarchive age days: 0.081

Porcentaje de aciertos en entrenamiento

Empleando el algoritmo Random Forest, el porcentaje de aciertos en entrenamiento fue también del 92.5 por ciento.

Resultados de la validación cruzada

Para realizar la validación cruzada se emplearon 10 subconjuntos de datos. El valor medio de todos los parámetros empleando el algoritmo Random Forest fue de 0.732, mientras que la desviación estándar fue de 0.017.

4.3.4. Gradient Boosting

Como se puede observar, Gradient Boosting es el algoritmo que mejor clasifica los datos. Quizás haya ayudado el hecho de que **en Gradient Boosting se ha empleado GridSearchCV.** Lo que permite hacer GridSearchCV es ejecutar el algoritmo utilizando distintos hiper parámetros, y devuelve el conjunto de hiper parámetros que mejor resultado da con ese algoritmo, según la métrica seleccionada.

Los datos obtenidos empleando este algoritmo fueron los siguientes:

Importancia según atributos

- SEMrush backlinks: 0.203
- SEMrush root domain backlinks: 0.072
- Alexa rank: 0.143
- SEMrush Rank: 0.085
- SEMrush SE Traffic: 0.09
- SEMrush SE Traffic price: 0.086
- Facebook likes number: 0.241
- Webarchive age days: 0.081

Porcentaje de aciertos en entrenamiento

Empleando el algoritmo Gradient Boosting, el porcentaje de aciertos en entrenamiento también fue del 92.5 por ciento.

Resultados de la validación cruzada

En esta ocasión, al igual que con el algoritmo Random Forest, se realizó validación cruzada empleando 10 subconjuntos de datos. El valor medio de todos los parámetros empleando el algoritmo Gradient Boosting fue de 0.755, mientras que la desviación estándar fue de 0.014.

Como se puede comprobar, el algoritmo Gradient Boosting es el que obtiene el mayor porcentaje de acierto, logrando un 75,46 por ciento.

4.4. Procedimiento para clasificar con cada algoritmo

A continuación se describe el procedimiento para clasificar con cada algoritmo. Se emplea como ejemplo el algoritmo Random Forest.

En primer lugar, de los datos cargados en la primera parte en la variable features se seleccionan los parámetros que serán empleados para clasificar. En la variable target se almacenan los valores resultado para el conjunto de features, como se puede observar en la figura Código 4.7

```
########## CLASIFICACION DE DATOS USANDO TREES Y ENSEMBLE TREES
    ##########
# De los datos se seleccionan las caracteristicas y el objetivo en el entrenamiento del modelo
target = data["Top_ranked"].values
features = data[["SEMrush_backlinks", "SEMrush_root_domain_backlinks", "Alexa_rank", "SEMrush_
    Rank", "SEMrush_SE_Traffic", "SEMrush_SE_Traffic_price", "Facebook_likes_number", "Webarchive_
    age_days"]].values
```

Después, se crea el modelo a utilizar empleando la librería de sklearn. Puede observarse en la figura Código 4.8

```
########## Creación de Random Forest
rand_forest = RandomForestClassifier(n_estimators=100, max_depth=None, min_samples_split=2,
    random_state=0)
```

Utilizando ese modelo y los datos de features y target se entrena al mismo, figura Código 4.9

```
rand_forest.fit(features, target)
```

En este punto, ya se puede observar la importancia de cada parámetro en el modelo y el porcentaje de aciertos sobre el conjunto de entrenamiento, se imprimen esos valores en la consola, como se puede observar en la figura Código 4.10

Como ya se ha comentado, **con Gradient Boosting se obtuvo el mejor resultado, que fue de 75 por ciento en la validación cruzada.** En la figura Código 4.11 se puede observar donde se realiza la validación cruzada.

```
# Imprimir importancia de atributos y resultado de entrenamiento
print("Importancia por atributos de Random Forest: ")
print("SEMrush backlinks -- SEMrush root domain backlinks -- Alexa rank -- SEMrush Rank --
    SEMrush SE Traffic -- SEMrush SE Traffic price -- Facebook likes number -- Webarchive age
    days ")
print(rand_forest.feature_importances_)
print('')
print("Porcentaje de aciertos en entrenamiento de Random Forest")
print(rand_forest.score(features, target))
print('')
```

```
scores = cross_val_score(grad_boost, features, target, cv=10)
print("Resultados de la validación cruzada (usando 10 subconjuntos de datos) con Gradient Boost: ")
print(scores)
print("Valor medio y desviación estándar")
print('{0} +- {1}'.format(scores.mean(), scores.std()))
print('')
```

4.5. Comparativa entre algoritmos

En las tablas 4.1 4.2 y 4.3 se puede observar la salida de los algoritmos mencionados, pudiendo apreciar en ella el porcentaje de aciertos en entrenamiento para cada algoritmo (empleando validación cruzada con 10 subconjuntos de datos) y la **relevancia (llamada en la tabla como "Rel")** de cada atributo en cada algoritmo (obviamente, cuanto mayor es la relevancia y/o es mayor en casi todos los algoritmos, se puede deducir que se traduce en una mayor relevancia de dicho parámetro). **El rango de cada relevancia es de 0 a 1** debido a la forma en que se normalizan las importancias (llamadas aquí relevancias) en los algoritmos de las tablas.

	Rel. Semrush Backlinks	Rel. Semrush Root Domain Backlinks	**Rel. Alexa Rank**
Tree	0.191	0.061	0.233
Extra Trees	0.223	0.067	0.057
Random Forest	0.203	0.072	0.143
Gradient Boosting	0.203	0.072	0.143

Tabla 4.1: El parámetro Rel. Semrush Backlinks representa los enlaces que recibe una URL en concreto, mientras que Rel. Semrush Root Domain Backlinks representa el número de enlaces que recibe el dominio de esa URL. Rel. Alexa Rank representa el ranking en Alexa. En negrita se destacan los parámetros más representativos.

Se pueden apreciar los valores más elevados destacados en negrita. Gradient Boosting (Descenso por Gradiente) es el algoritmo que más acierta clasificando, mientras que los parámetros Semrush Backlinks, Alexa Rank y Facebook Likes Number fueron los que más importancia demostraron tener. Dicho en otras palabras, son parámetros decisivos a la hora de clasificar una URL en las 3 primeras

	Rel. Semrush Rank	Rel. Semrush SE Traffic	Rel. Semrush SE Traffic Price
Tree	0.047	0.044	0.076
Extra Trees	0.059	0.127	0.119
Random Forest	0.085	0.09	0.086
Gradient Boosting	0.085	0.09	0.086

Tabla 4.2: El parámetro Rel. Semrush Rank representa el ranking de una URL en la herramienta Semrush, mientras que Rel. Semrush SE Traffic representa el tráfico que recibe esa URL. Rel. Semrush SE Traffic Price representa la ganancia estimada teniendo en cuenta el tráfico de esa URL.

	Rel. Facebook Likes Number	Rel. Webarchive Age Days	Val. Medio Entrenamiento
Tree	0.214	0.133	67.9 %
Extra Trees	0.267	0.081	71.8 %
Random Forest	0.241	0.081	73.2 %
Gradient Boosting	0.241	0.081	**75.5 %**

Tabla 4.3: El parámetro Rel. Facebook Likes Number representa el número de Likes en Facebook que recibe una URL en concreto. Rel. Webarchive Age Days representa el número de días que lleva activa una URL. Finalmente se obtiene el valor medio de entrenamiento (Val. Medio Entrenamiento). En negrita se destacan los parámetros y valores más representativos. Se puede apreciar en la tabla que el mejor valor en entrenamiento es obtenido con el algoritmo Gradient Boosting.

posiciones del buscador de Google, en la zona “Top Ranked”.

4.6. Acciones a realizar en base al análisis realizado

Una vez realizado el análisis entre los diferentes algoritmos, se puede determinar los parámetros que serán más relevantes a la hora de posicionar una web.

Esto significa que es necesario actuar sobre esos parámetros para lograr un mejor posicionamiento en el buscador de Google.

Tomando como ejemplo el análisis anteriormente realizado sobre el diccionario de 1000 palabras clave, se puede concluir que **hay que actuar sobre los parámetros más relevantes, que son Rel. Semrush Backlinks, Rel. Alexa Rank y Rel. Facebook Likes Number**.

Tener que actuar sobre estos parámetros significa que hay que aumentar el número de backlinks a la URL que se desea posicionar, ascender en el ranking de Alexa (esto ya se logra simplemente aumentando el número de backlinks) y logrando un mayor número de likes en facebook para la URL a posicionar.

Tomando esta sistemática, se le puede proporcionar al algoritmo del buscador de Google las señales positivas (parámetros) que necesita para tomar en cuenta una URL en concreto.

5

Conclusiones y trabajo futuro

5.1. Conclusiones

Como conclusión final, según lo visto en los distintos algoritmos utilizados, los parámetros que más importancia tienen cuando se clasifican los datos son:

1.– SEMrush backlinks (22 por ciento con Extra Trees, 20 por ciento Random Forest y Gradient Boosting)

2.– Alexa rank (14 por ciento con Random Forest y Gradient Boosting)

3.– Facebook likes (26 por ciento con Extra Trees y 24 por ciento Random Forest y Gradient Boosting)

Estos resultados efectivamente coinciden con la opinión ampliamente extendida en la comunidad SEO, que enfatiza la gran importancia que tiene para el ranking de una web el tener una gran cantidad de enlaces (SEMrush backlinks) y el número de likes en Facebook (Facebook likes), ya que el buscador de Google le otorga una gran importancia al factor social y a la experiencia de usuario de una web.

Por tanto podemos afirmar que:

- Se han determinado cuales son los factores principales que intervienen en el posicionamiento que determinan el mejor posicionamiento de las URL que mejor posicionan en las SERPS del buscador de Google.
- Se ha buscado y encontrado la mejor herramienta para obtener los parámetros de las URL posicionadas en el buscador de Google para el diccionario de palabras clave semilla.
- Se ha diseñado un sistema para la extracción de los datos relevantes que aporta la herramienta SEOquake.
- Aunque Google siga actualizando su familia de algoritmos de posicionamiento web en su buscador, la sistemática diseñada y utilizada en este trabajo permite establecer cuales bajo cualquier cambio los factores determinantes analizando los datos volcados por la herramienta SEOquake introduciendo el diccionario de palabras clave semilla.

Todo esto permite afirmar que los objetivos perseguidos por este trabajo se han conseguido completamente.

5.2. Trabajo futuro

Utilizando los algoritmos descritos en el capítulo 4, aunque se ha logrado que el modelo aprenda utilizando los datos, se puede lograr en un futuro que el algoritmo clasifique de forma más precisa a partir de los parámetros.

Para lograrlo, probablemente falten más datos por palabra (en cantidad) y sobre todo falten más parámetros en cada registro. Otra cosa que ayudaría a la clasificación sería conseguir otros parámetros de cada URL que permitan una mejor clasificación de datos. Otras mejoras adicionales podrían ser:

1.– **Ampliar la cantidad de parámetros obtenidos en la herramienta SEOquake**: si la herramienta devuelve mayor cantidad de parámetros tales como número de veces compartida la URL en Facebook, velocidad de carga de la URL, etc. se puede brindar un análisis más completo, ya que al contar con mayor cantidad de parámetros, aumentan las probabilidades de obtener una fórmula fiel a la realidad.

2.– **Ampliar o mejorar el diccionario de palabras empleado**: se puede ampliar con más palabras, máximo 10.000, o modificar en base a ciertas investigaciones. Otra opción más interesante es emplear entre 5 a 10 diccionarios.

Empleando una de las dos formas de mejorar el análisis en Python o ambas a la vez, se pueden mejorar sin duda los resultados obtenidos, ya que estaríamos realizando un análisis más exhaustivo sobre una mayor cantidad de parámetros y palabras semilla. Se ha decidido partir de un diccionario y los parámetros mostrados porque son lo suficientemente ilustrativos como para poder establecer una sistemática de análisis de posicionamiento web válida. Además, hoy en día no existen herramientas más complejas y con mayor cantidad de parámetros que SEOquake a la hora de analizar una página del buscador de Google en base a cierta palabra clave. Quizás alguna sea más completa que SEOquake, pero son de uso profesional, así como esta, pero de coste económico realmente prohibitivo.

Bibliografía

[1] mkcheck, “Evolución del algoritmo de Google en SEO - MKCheck.” url: https://www.mkcheck.com/evolucion-del-algoritmo-de-google-en-seo/, 2016.

[2] Dean Romero, “Esto son los 200 factores que Google tiene en cuenta para posicionar tu página | Quondos.” url: https://www.quondos.com/factores-seo-posicionamiento-web-google/, 2015.

[3] JoseLab, “Analizar competencia de palabras clave.” url: https://joselab.com/analizar-competencia-de-palabras-clave/, 2015.

[4] Arturo Cenzano, “El hablante de espanol utiliza cada vez menos palabras | Sentidos | Cinco Dias.” url: https://cincodias.elpais.com/cincodias/2005/06/06/sentidos/1118024839_850215.html, 2005.

[5] SEOquake, “A Powerful SEO Toolbox for your Browser – SEOquake.” url: https://www.seoquake.com/index.html, 2019.

[6] Thais Balaguero, “Los 4 principales usos del Big Data | Deusto Formacion.” url: https://www.deustoformacion.com/blog/gestion-empresas/4-principales-usos-big-data, 2017.

[7] Redaccion Digital Tech Institute, “Los 8 mejores lenguajes de programacion para IA | Digital Tech Institute.” url: https://www.deustoformacion.com/blog/gestion-empresas/4-principales-usos-big-data, 2018.

[8] _MACHINE LEARNING, “Los 10 Algoritmos esenciales en Machine Learning - Raona.” url: https://www.raona.com/los-10-algoritmos-esenciales-machine-learning/, 2017.

[9] Juan Zambrano, “Aprendizaje supervisado o no supervisado – Juan Zambrano – Medium.” url: https://medium.com/@juanzambrano/aprendizaje-supervisado-o-no-supervisado-39ccf1fd6e7b, 2018.

[10] Marcelo R. Ferreyra, “Data Mining basado en Teoría de la Información: ¿Qué es el ruido?.” url: http://powerhousedm.blogspot.com/2007/10/qu-es-el-ruido.html, 2007.

[11] Andrea Lehr, “White vs. Black Hat SEO: What is the Difference? | Search Engine Journal.” url: https://www.searchenginejournal.com/white-vs-black-hat-seo-what-is-the-difference/183088/, 2017.

[12] Luis M. Villanueva, “SEMrush. Manual completo SEMRUSH ESPAÑOL..” url: https://luismvillanueva.com/seo/semrush.html, 2015.

[13] Javier Marcilla, “SERPWoo es la solución definitiva para dominar tu Nicho.” url: https://ninjaseo.es/serpwoo-rank-tracker-monitoriza-nichos/, 2018.

[14] Jeff Desjardins, “How Google retains more than 90% of market share - Business Insider.” url: https://www.businessinsider.com/how-google-retains-more-than-90-of-market-share-2018-4?IR=T, 2018.

[15] Carlos De La Ossa, “Las 5 cosas que nadie le dice sobre Google AdWords.” url: https://delaossa.co/blog/16-contenidos/blog/publicidad-digital/163-las-5-cosas-que-nadie-le-

dice-sobre-google-adwords, 2016.

[16] WordStream, “What Is Google AdWords? How the AdWords Auction Works.” url: https://www.wordstream.com/articles/what-is-google-adwords, 2018.

[17] Servando Silva, “6 months after my Backlink Experiment - What works for Google and what doesn’t - Stream SEO.” url: https://stream-seo.com/6-months-backlink-experiment/, 2017.

[18] Emilio García, “Experimentos Linkbuilding: Probando la eficacia de los enlaces en prensa [Parte I].” url: https://www.unancor.com/blog/experimentos-reales-linkbuilding-1a-parte/, 2018.

[19] Anuja Nagpal, “Decision Tree Ensembles- Bagging and Boosting – Towards Data Science.” url: https://towardsdatascience.com/decision-tree-ensembles-bagging-and-boosting-266a8ba60fd9, 2017.

Definiciones

algoritmo de posicionamiento Familia de algoritmos cuyo único objetivo es ordenar las URL de un buscador con el fin de mostrar en las primeras posiciones las más relevantes.

aprendizaje automático También llamado machine learning, subcampo de las ciencias de la computación y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que los computadores aprendan.

big data Concepto que hace referencia a conjuntos de datos tan grandes y complejos como para que aplicaciones informáticas tradicionales de procesamiento de datos puedan tratarlos adecuadamente.

blackhat Uso de técnicas para engañar a un buscador a fin de obtener resultados provechosos para quien lo hace en los resultados de búsqueda.

C Lenguaje de programación ampliamente extendido desarrollado entre 1969 y 1972.

competencia En este contexto, nivel de dificultad para que una URL alcance las posiciones más altas en los rankings de un buscador.

conjunto de entrenamiento Creado a partir de una fuente de objetos clasificados de determinada manera, debe contener una lista de objetos con tipos conocidos.

experiencia de usuario Conjunto de factores y elementos relativos a la interacción del usuario, con un entorno o dispositivo concretos, cuyo resultado es la generación de una percepción positiva o negativa de dicho servicio, producto o dispositivo.

expresiones regulares Secuencia de caracteres que forman un patrón de búsqueda, principalmente utilizada para la búsqueda de patrones de cadenas de caracteres u operaciones de sustituciones.

extensión Pequeños programas instalados en un navegador web que añaden o mejoran funciones del navegador.

factores de posicionamiento También llamados parámetros, son valores que los algoritmos de posicionamiento tienen en cuenta a la hora de posicionar una web.

Flex Analizador léxico utilizado con expresiones regulares.

Google Panda Cambio en el algoritmo de clasificación de los resultados de búsqueda de Google.

Google Penguin Nombre en clave para una actualización del algoritmo de Google que se anunció por primera vez el 24 de abril de 2012.

heurística Propone estrategias que guían el descubrimiento.

hiper parámetros Parámetros del algoritmo de aprendizaje, tales como el parámetro learning_rate o la cantidad de estimadores que tendrá el ensembling.

posicionar Lograr que cualquier algoritmo de posicionamiento coloque en los primeros lugares de la lista de resultados a una página web.

proxy También llamado servidor proxy, es un agente o sustituto autorizado para actuar en nombre de otra persona o un documento que lo autoriza a hacerlo.

python Lenguaje de programación. Ha sido el lenguaje de programación elegido en este TFG para crear el fichero llamado modelo.py.

query También llamada palabra clave en este ámbito, son las consultas o preguntas realizadas al buscador de Google por sus usuarios.

ruido En este contexto, datos que interfieren en un análisis, distorsionando los resultados obtenidos.

scikit learn También llamado sklearn, es una biblioteca de aprendizaje automático de software libre para el lenguaje de programación Python.

scraper Aplicación que extrae información de sitios web.

semilla En este contexto, un fichero semilla es un fichero del que se extraen ficheros mas elaborados.

sistemática Sistema o método con que se clasifica algo.

spam Mensajes o peticiones no solicitadas, no deseadas y sumamente molestas.

webmaster Persona que posee y administra varias páginas web, usualmente con el objetivo de obtener beneficio económico.

whitehat Uso de técnicas legítimas aprobadas por las directrices para webmasters de Google a fin de obtener resultados provechosos para quien lo hace en los resultados de búsqueda.

Acrónimos

API Application Programming Interface.

CPC Cost Per Click.

CREA Corpus de Referencia del Español Actual.

NaN Not a Number.

RAE Real Academia Española.

SEO Search Engine Optimization.

SERPS Search Engine Results Pages.

TFG Trabajo de Fin de Grado.

URL Uniform Resource Locator.